BUNDESREPUBLIK DEUTSCHLAND

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen:	102 29 817.3
Anmeldetag:	28. Juni 2002
Anmelder/Inhaber:	Robert Bosch GmbH, Stuttgart/DE
Bezeichnung:	Verfahren und Vorrichtung zum Ablegen eines Computerprogramms in einen Programmspeicher eines Steuergeräts
IPC:	G 06 F 11/07

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 1. April 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Faust

A 9161
02/00
EDV-L

R.302905

28.06.2002

Robert Bosch GmbH, 70442 Stuttgart

Verfahren und Vorrichtung zum Ablegen eines Computerprogramms in einen Programmspeicher eines Steuergeräts

Stand der Technik

Die vorliegende Erfindung betrifft ein Verfahren zum Ablegen eines Computerprogramms in einen Programmspeicher eines Steuergeräts, wobei das Computerprogramm nach vorgebbaren Regeln in bestimmten Speicherbereichen des Programmspeichers abgelegt wird.

Die Erfindung betrifft außerdem eine Vorrichtung zum Ablegen eines Computerprogramms in einen Programmspeicher eines Steuergeräts, wobei die Vorrichtung erste Mittel zum Ablegen des Computerprogramms nach vorgebbaren Regeln in bestimmte Speicherbereiche des Programmspeichers aufweist.

Schließlich betrifft die vorliegende Erfindung ein Steuergerät mit einem Rechengerät, insbesondere einem Mikroprozessor, und einem Programmspeicher, auf dem ein Computerprogramm nach vorgebbaren Regeln in bestimmten Speicherbereichen des Programmspeichers abgelegt ist.

Aus dem Stand der Technik ist es bspw. bei Kraftfahrzeug-Steuergeräten bekannt, den Programmcode eines Computerprogramms für ein Rechengerät, insbesondere für einen Mikroprozessor oder für eine CPU (Central Processing Unit), und für eventuell vorhandene Co-Prozessoren entsprechend einem für das jeweilige Steuergerät gültigen Adressmapping zu lokatieren und in einem Programmspeicher des Steuergeräts abzulegen. Unter Lokatieren wird das Zuweisen bestimmter Teile des Computerprogramms, sog. Programmsegmente, zu bestimmten Speicherbereichen des Programmspeichers verstanden.

Nach dem Stand der Technik erfolgt das Lokatieren und Ablegen des Programmcodes in dem Programmspeicher nach vorgebbaren Regeln, die insbesondere die nachfolgenden Sachverhalte berücksichtigen:

- Programmsegmente, die häufig aufgerufen werden, werden in Speicherbereiche lokatiert, die eine schnelle Programmausführung, d. h. eine schnelle Abarbeitung der Programmsegmente auf dem Mikroprozessor oder der CPU ermöglichen. Diese Programmsegmente (bspw. der Programmcode schneller Zeitraster) können in einem internen Flash-Speicher des Steuergeräts abgelegt werden.

- Die Zugriffsmöglichkeiten auf den Programmspeicher bei bestimmten hardwarebedingten Systemzuständen. So kann es bspw. vorkommen, dass bei Unterspannung nicht auf den internen Flash-Speicher zugegriffen werden kann. Um diesem Systemzustand Rechnung zu tragen, werden die Programmsegmente, auf die trotz Unterspannung sicher zugegriffen werden soll, in einen externen Flash-

Speicher lokatiert und dort abgelegt.

Im Rahmen der vorliegenden Erfindung wird unter internem Flash-Speicher derjenige Flashbereich bezeichnet, der sich innerhalb des CPU-Gehäuses befindet. Mit externem Flash-Speicher ist dagegen ein separater IC (Integrated Circuit)-Baustein bezeichnet, auf den über einen externen Bus von der CPU aus zugegriffen wird.

Die Lokatierung erfolgt nach dem Assemblieren, Compilieren und Linken des Programmcodes und bevor das Computerprogramm in den Programmspeicher des Steuergeräts abgelegt wird. Insgesamt führt das aus dem Stand der Technik bekannte Verfahren zum Ablegen eines Computerprogramms in einem Programmspeicher eines Steuergeräts dazu, dass Programmsegmente auf unterschiedliche, nicht zusammenhängende Adressbereiche des Programmspeichers verteilt werden.

Beim Abarbeiten des in dem Programmspeicher abgelegten Computerprogramms auf einem Rechengerät, insbesondere auf einem Mikroprozessor oder einer CPU, kann es aus unterschiedlichen Gründen dazu kommen, dass das Computerprogramm in ungenutzte Speicherbereiche des Programmspeichers, in denen kein Programmcode abgelegt ist, springt. In den ungenutzten Speicherbereichen ist nach dem Stand der Technik kein definierter Programmcode abgelegt. Nach einem Sprung in den ungenutzten Speicherbereich des Programmspeichers wird folglich dieser undefinierte Programmcode abgearbeitet. Dadurch kann das Steuergerät in einen undefinierten und damit irregulären Zustand gelangen.

Ursachen für einen Sprung des Computerprogramms in den ungenutzten Speicherbereich des Programmspeichers können

innere und äußere Einflüsse, bspw. Bitkipper in dem Flash-Speicher oder in einem RAM (Random-Access-Memory), Auswirkungen von überhöhter EMV (Elektromagnetische Verträglichkeit)-Strahlung oder schlummernde Programmierfehler sein.

Aus dem Stand der Technik sind des weiteren verschiedene Mechanismen bekannt, um einen irregulären Zustand des Steuergeräts zu erkennen und das System einerseits in einen sicheren Zustand zu überführen und andererseits die Funktionalität des Steuergerätes wieder zu gewährleisten. Diese bekannten Mechanismen umfassen bspw.:

- einen internen Controller-Watchdog;
- eine Überwachung von Zeitrastern;
- ein Zwei-Rechner-Konzept;
- eine Überwachung des Programmablaufs auf Plausibilität;
- eine Checksummenprüfung.

Durch die beispielhaft aufgezählten Mechanismen wird versucht, Bitkipper in dem Flash-Speicher oder in dem RAM, Einflüsse durch elektromagnetische Einstrahlung (EMV) oder nichtplausible Zustände wie z.B. schlummernde Programmierfehler (bspw. Sprünge über falsch berechnete Pointer) direkt oder indirekt zu erkennen, das Steuergerät in einen sicheren Zustand zu überführen und die Funktionalität des Steuergerätes wieder herzustellen. Durch das frühzeitige Erkennen von irregulären oder undefinierten Zuständen des Steuergerätes soll die Robustheit des Systems

erhöht werden. Durch eine rasche Wiederherstellung der Funktionalität des Steuergerätes soll die Verfügbarkeit des Systems verbessert werden.

Aus der DE 100 18 859 A1 ist es bekannt, beim Auftreten einer Fehlfunktion einer Einrichtung zum Messen, Steuern und Regeln (MSR) ein Überwachungssystem für die MSR-Einrichtung nicht sofort, sondern erst nach mehrfachem Auftreten einer Fehlfunktion in einen sicheren Zustand zu überführen. Bei jedem Auftreten einer Fehlfunktion wird der Zählerstand eines Zählers erhöht. Überschreitet der Zählerstand einen vorgebbaren Grenzwert, geht das Überwachungssystem in den sicheren Zustand über.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, einen weiteren Mechanismus zu schaffen, durch den undefinierte oder irreguläre Zustände des Steuergerätes erkannt, sowie das Steuergerät in einen sicheren Zustand überführt und die Funktionalität des Steuergerätes wieder hergestellt werden kann.

Zur Lösung dieser Aufgabe schlägt die vorliegende Erfindung ausgehend von dem Verfahren der eingangs genannten Art vor, dass in ungenutzten Speicherbereichen des Programmspeichers, in denen das Computerprogramm nicht abgelegt wird, vorgebbare Informationen abgelegt werden, durch die das Steuergerät in einen definierten Zustand überführt wird.

Vorteile der Erfindung

Ein wesentlicher Aspekt der vorliegenden Erfindung besteht somit darin, in den ungenutzten Speicherbereichen des Programmspeichers statt des undefinierten Programmcodes

vorgebbare Informationen, vorzugsweise einen bestimmten Programmcode abzulegen, durch den das Steuergerät in einen definierten Zustand überführt wird. Zusätzlich kann durch die vorgebbaren Informationen auch die Funktionalität des Steuergerätes wieder hergestellt werden.

Die Erfindung betrifft einen Mechanismus, der verhindert, dass das Rechengerät, insbesondere der Mikroprozessor oder die CPU, durch Speicherbereiche läuft, die in dem entsprechenden Stand des Computerprogramms eigentlich ungenutzt sind und folglich für die Programmausführung nicht verwendet werden dürfen. Falls das Rechengerät in diese Speicherbereiche verzweigt, liegt auf jeden Fall ein nicht plausibler Zustand vor. Um den weiteren Ablauf des Computerprogramms durch diesen ungenutzten Speicherbereich hindurch dennoch kontrollieren zu können und eine zufällige Rückkehr in vorhandenen Programmcode des Computerprogramms auszuschließen, soll dieser Speicherbereich zumindest teilweise mit einem speziellen Programmcode aufgefüllt werden, durch den das Rechengerät gezielt in einen definierten Zustand überführt wird. Vorzugsweise wird das Rechengerät durch den speziellen Programmcode veranlasst den ungenutzten Speicherbereich zu verlassen.

Um eine Diagnose über die Ursache des Fehlers, der zu dem Sprung in den ungenutzten Speicherbereich des Programmspeichers geführt hat, zu erhalten, können zusätzlich Vergangenheitsinformationen in einer Interrupt-Service-Routine oder in einer Fehlerbehandlungs-Routine abgespeichert werden.

Sobald in einen ungenutzten Speicherbereich gesprungen wird bzw. sobald Programmbefehle aus diesem Speicherbereich ausgeführt werden, wird dies erkannt. Das Steuergerät wird

sofort oder erst nach einer speziellen Fehlerbehandlung zurückgesetzt. Durch einen anschließenden Hochlauf des Steuergeräteprogramms wird das System wieder in einen definierten, funktionsfähigen Zustand versetzt. Nach dem Hochfahren des Steuergerätes kann dann, sofern kein dauerhafter Fehler vorliegt, wieder mit der normalen Funktionalität des Steuergerätes fortgefahren werden.

Der erfindungsgemäß vorgeschlagene Mechanismus bietet eine Absicherungsmaßnahme gegen Programmausführung in den ungenutzten Speicherbereichen eines Programmspeichers, die in dem entsprechenden Stand des Computerprogramms nicht verwendet werden dürfen. Durch die vorliegende Erfindung wird die Robustheit eines Computerprogramms für ein Steuergerät erhöht und die Verfügbarkeit des Steuergeräts entscheidend verbessert. Das Auftreten eines nichtplausiblen bzw. irregulären Zustands wird sofort erkannt. Darüber hinaus ist eine nachträgliche Implementierung in der Software aller Steuergeräte möglich. Der erfindungsgemäß vorgeschlagene Mechanismus kann einfach und schnell in der Steuergerätesoftware implementiert werden. Da eine Erweiterung des Programmcodes des Computerprogramms nicht notwendig ist, fällt kein zusätzlicher Aufwand und fallen keine zusätzlichen Kosten für die Realisierung der vorliegenden Erfindung an.

Es ist theoretisch denkbar, dass ein Rechengerät des Steuergerätes, insbesondere ein Mikroprozessor oder eine CPU, während der Abarbeitung des Computerprogramms in Speicherbereiche des Flash-Speichers springt bzw. Speicherbereiche des Flash-Speichers durchläuft, die zwar physikalisch vorhanden sind, die aber in dem aktuellen Stand des Computerprogramms ungenutzt sind. Falls das Rechengerät fälschlicherweise in diese ungenutzten

Speicherbereiche springt und/oder in diesen ungenutzten Speicherbereichen Programmcode ausliest, kann das Steuergerät in einen irregulären oder undefinierten Zustand gelangen. Das Rechengerät wird versuchen, den aus den ungenutzten Speicherbereichen ausgelesenen Programmcode auszuführen. Falls der Programmcode keinen Sprung beinhaltet, wird das Rechengerät linear die gelesenen Programmbefehle ausführen und mit großer Wahrscheinlichkeit irgendwann in einen Speicherbereich mit regulärem Programmcode des Computerprogramms wieder hineinlaufen. Das Systemverhalten während der Ausführung der Programmbefehle aus dem ungenutzten Speicherbereich als auch das Systemverhalten nach einem Übergang in den tatsächlich genutzten Speicherbereich des Flash-Speichers ist nicht definiert bzw. nicht vorhersehbar und muss deshalb vermieden werden. Besonders kritisch ist dieses Verhalten, wenn es in einem Zeitraster geschieht, das selbst nicht auf Plausibilität kontrolliert wird.

Mit der vorliegenden Erfindung kann ein solcher undefinierter bzw. irregulärer Zustand des Steuergerätes sofort wieder abgebrochen werden, wodurch die Robustheit des Steuergeräts erhöht wird. Außerdem kann durch geeignete Wahl der vorgebbaren Informationen das Gesamtsystem schnell wieder in einen sicheren Zustand überführt und die Funktionalität des Steuergerätes wieder hergestellt werden, wodurch die Verfügbarkeit des Steuergeräts entscheidend verbessert wird.

Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass das Steuergerät durch Abarbeiten der vorgebbaren Informationen auf einem Rechengerät, insbesondere auf einem Mikroprozessor oder auf einer CPU (Central Processing Unit), des Steuergeräts

zurückgesetzt wird. Durch die in dem ungenutzten Speicherbereich abgelegten Informationen wird also gezielt ein Reset des Steuergerätes ausgelöst. Der Reset wird vorzugsweise durch einen entsprechenden Programmbefehl bewirkt (sog. Software-Reset) oder durch einen in der Recheneinheit nicht vorhandenen und damit verbotenen Befehlscode (sog. Illegal Opcode) oder bspw. durch einen Befehl „trap unconditionally" für die Verzweigung in eine Interrupt-Service Routine und/oder eine Fehlerbehandlungs-Routine.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass durch Abarbeiten der vorgebbaren Informationen auf einem Rechengerät, insbesondere auf einem Mikroprozessor oder einer CPU, des Steuergeräts eine Interrupt-Service-Routine aufgerufen wird. Bei einem Interrupt wird ein laufendes Computerprogramm eines Prozessors zugunsten eines dringenderen Programms unterbrochen. Bei Auftreten eines Interrupts rettet der Prozessor alle für die Weiterarbeit am laufenden Computerprogramm notwendigen Daten in einen speziellen Speicherbereich, dem sogenannten Stapelspeicher oder Stack. Nach dem Abarbeiten der Interrupt-Service-Routine setzt der Prozessor dann das laufende Computerprogramm fort. Im Rahmen des Programms, das durch die Interrupt-Service-Routine aufgerufen wird, kann das System in einen sicheren Zustand überführt und die Funktionalität des Steuergeräts wieder hergestellt werden.

Vorteilhafterweise wird durch Abarbeiten der vorgebbaren Informationen auf einem Rechengerät, insbesondere auf einem Mikroprozessor oder einer CPU, des Steuergeräts eine Fehlerbehandlungs-Routine aufgerufen.

Vorzugsweise wird das Steuergerät am Ende der Interrupt-Service-Routine und/oder am Ende der Fehlerbehandlungs-Routine zurückgesetzt. Das Zurücksetzen des Steuergeräts kann bspw. durch einen entsprechenden Programmbefehl bewirkt werden (sog. Software-Reset) oder durch einen in der Recheneinheit nicht vorhandenen und damit verbotenen Befehlscode (sog. Illegal Opcode) oder bspw. durch einen Befehl „trap unconditionally" für die Verzweigung in eine Interrupt-Service Routine und/oder eine Fehlerbehandlungs-Routine. Der Reset hat einen Hochlauf des Steuergeräteprogramms zur Folge hat. Sowohl im Rahmen der Interrupt-Service-Routine als auch im Rahmen der Fehlerbehandlungs-Routine können Informationen über den genauen Ort des Auftretens und die Vergangenheit festgehalten werden (z.B. eine Rücksprungadresse in das Computerprogramm). Aus diesen Informationen können Rückschlüsse (z.B. auf die Häufigkeit des Auftretens eines Fehlers) gezogen werden.

Die genaue Umsetzung der vorliegenden Erfindung hängt einerseits von dem verwendeten Rechengerät, insbesondere von dem Typ des verwendeten Mikroprozessors oder der verwendeten CPU ab. Verschiedene Rechengeräte unterscheiden sich bspw. durch den verwendeten Befehlssatz, der auf dem Rechengerät abgearbeitet wird. Andererseits hängt die genaue Umsetzung der vorliegenden Erfindung von dem gewünschten Umfang der Funktionalität des Mechanismus, d. h. von der gewünschten "Intelligenz" der Erkennung, der Überführung in einen sicheren Zustand und der Wiederherstellung der Funktionsfähigkeit des Steuergerätes, ab.

Es ist denkbar, die vorgebbaren Informationen lediglich in ausgewählten ungenutzten Speicherbereichen des

Programmspeichers abzulegen. Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird jedoch vorgeschlagen, dass die vorgebbaren Informationen in allen ungenutzten Speicherbereichen des Programmspeichers abgelegt werden.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass wenigstens ein ungenutzter Speicherbereich des Programmspeichers mit den vorgebbaren Informationen vollständig aufgefüllt wird. Es ist denkbar, lediglich ausgewählte oder aber alle ungenutzten Speicherbereiche des Programmspeichers vollständig mit den vorgebbaren Informationen aufzufüllen.

Gemäß einer anderen bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass die vorgebbaren Informationen in vorgebbaren Abständen in wenigstens einem ungenutzten Speicherbereich des Programmspeichers abgelegt werden, wobei der Teil des ungenutzten Speicherbereichs, in dem die vorgebbaren Informationen nicht abgelegt werden, keine Sprünge und keine Endlosschleifen bewirkt. Die vorgebbaren Informationen werden vorzugsweise in regelmäßigen Abständen in dem wenigstens einen ungenutzten Speicherbereich des Programmspeichers abgelegt.

Gemäß noch einer anderen bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass die vorgebbaren Informationen lediglich am Ende wenigstens eines ungenutzten Speicherbereiches des Programmspeichers abgelegt werden. Dabei muss jedoch sichergestellt sein, dass der Teil des ungenutzten Speicherbereiches, in dem keine vorgebbaren Informationen abgelegt werden, keine Sprünge und keine Endlosschleifen bewirkt.

Als eine weitere Lösung der Aufgabe der vorliegenden Erfindung wird ausgehend von der Vorrichtung der eingangs genannten Art vorgeschlagen, dass die Vorrichtung zweite Mittel zum Ablegen von vorgebbaren Informationen, welche das Steuergerät in einen definierten Zustand überführen, in ungenutzte Speicherbereiche des Programmspeichers aufweist, in denen die ersten Mittel das Computerprogramm nicht abgelegt haben.

Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass die zweiten Mittel als ein Hexadezimal-Editor ausgebildet sind. Durch den Hexadezimal-Editor können die ungenutzten Speicherbereiche, die in dem entsprechenden Stand des Computerprogramms nicht genutzt werden, mit speziellem Hexadezimal-Code gefüllt werden. Das Befüllen des nichtgenutzten Programmspeichers erfolgt im Verlaufe der Herstellung des Standes des Computerprogramms.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass die Vorrichtung Mittel zur Ausführung des erfindungsgemäßen Verfahrens aufweist.

Als noch eine weitere Lösung der Aufgabe der vorliegenden Erfindung wird ausgehend von dem Steuergerät der eingangs genannten Art vorgeschlagen, dass in ungenutzten Speicherbereichen des Programmspeichers, in denen das Computerprogramm nicht abgelegt ist, vorgebbare Informationen abgelegt sind, durch die das Steuergerät in einen definierten Zustand überführbar ist.

Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass die vorgebbaren

Informationen nach dem erfindungsgemäßen Verfahren in den ungenutzten Speicherbereichen des Programmspeichers abgelegt sind.

Zeichnungen

Weitere Merkmale, Anwendungsmöglichkeiten und Vorteile der Erfindung ergeben sich aus der nachfolgenden Beschreibung von Ausführungsbeispielen der Erfindung, die in der Zeichnung dargestellt sind. Dabei bilden alle beschriebenen oder dargestellten Merkmale für sich oder in beliebiger Kombination den Gegenstand der Erfindung, unabhängig von ihrer Zusammenfassung in den Patentansprüchen oder deren Rückbeziehung sowie unabhängig von ihrer Formulierung bzw. Darstellung in der Beschreibung bzw. in der Zeichnung. Es zeigen:

Figur 1 ein Ablaufdiagramm eines erfindungsgemäßen Verfahrens gemäß einer bevorzugten Ausführungsform; und

Figur 2 ein erfindungsgemäßes Steuergerät gemäß einer bevorzugten Ausführungsform.

Beschreibung der Ausführungsbeispiele

In Figur 1 ist ein Ablauf des erfindungsgemäßen Verfahrens gemäß einer bevorzugten Ausführungsform dargestellt. Das erfindungsgemäße Verfahren beginnt in einem Funktionsblock 1. In einem Funktionsblock 2 wird der Programmcode für ein bestimmtes Computerprogramm erstellt. Das Computerprogramm dient bspw. zur Steuerung und/oder Regelung einer bestimmten Funktionalität in einem Kraftfahrzeug. In einem Funktionsblock 3 wird das erstellte Computerprogramm

assembliert, kompiliert und gelinkt. Anschließend erfolgt in einem Funktionsblock 4 die Lokatierung des Programmcodes. Im Rahmen der Lokatierung wird das Computerprogramm nach vorgebbaren Regeln bestimmten Speicherbereichen eines Programmspeichers zugeordnet. Beim Lokatieren des Programmcodes werden bspw. folgende Sachverhalte berücksichtigt:

- Teile des Computerprogramms, sogenannte Programmsegmente, die häufig aufgerufen werden (bspw. der Programmcode schneller Zeitraster), werden in Speicherbereichen lokatiert, die eine schnelle Programmausführung ermöglichen, bspw. in einem internen Flash-Speicher.

- Programmsegmente, auf die trotz bestimmter hardwarebedingter Systemzustände mit Sicherheit noch zugegriffen werden soll, werden in entsprechende Speicherbereiche lokatiert, auf die auch in diesem Systemzuständen noch sicher zugegriffen werden kann. Falls bspw. bei Unterspannung ein Zugriff auf einen internen Flash-Speicher nicht möglich ist, werden die entsprechenden Programmsegmente in externe Flash-Speicher lokatiert. Durch das Lokatieren wird das Computerprogramm auf unterschiedliche, nicht zusammenhängende Adressbereiche des Programmspeichers verteilt.

In einigen Speicherbereichen des Programmspeichers wird das Computerprogramm nicht abgelegt, diese Speicherbereiche bleiben also ungenutzt. Es ist theoretisch denkbar, dass ein Rechengerät, insbesondere ein Mikroprozessor oder eine CPU (Central Processing Unit), bei der Programmausführung in diese ungenutzten Speicherbereiche des Programmspeichers springt bzw. diese ungenutzten Speicherbereiche durchläuft.

Diese ungenutzten Speicherbereiche sind zwar physikalisch vorhanden, werden aber bei dem aktuellen Programmstand nicht genutzt und sind somit frei. Falls das Rechengerät fälschlicherweise in diese ungenutzten Speicherbereiche springt und/oder in diesen ungenutzten Speicherbereichen Programmcode ausliest, kann das Steuergerät in einen undefinierten und damit irregulären Zustand gelangen. Das Rechengerät versucht, den aus dem ungenutzten Speicherbereich ausgelesenen Programmcode auszuführen. Falls der Programmcode keinen Sprung beinhaltet, führt das Rechengerät die eingelesenen Programmbefehle linear aus und läuft mit großer Wahrscheinlichkeit in einen Speicherbereich hinein, in dem regulärer Programmcodes des Programms abgelegt ist.

Das Systemverhalten während der Abarbeitung von Programmbefehlen aus dem ungenutzten Speicherbereich und nach einem Übergang in den von dem Computerprogramm tatsächlich genutzten Speicherbereich des Programmspeichers ist nicht vorhersehbar und muss deshalb vermieden werden. Besonders kritisch ist dieses Verhalten, wenn es in einem Zeitraster geschieht, das selbst nicht auf Plausibilität kontrolliert wird.

Die vorliegende Erfindung schlägt einen weiteren Mechanismus vor, durch den dieser irreguläre Zustand sofort bei dessen Auftreten abgebrochen, das Gesamtsystem in einen sicheren Zustand überführt und die Funktionalität des Steuergeräts wieder hergestellt wird. Dazu werden bei dem erfindungsgemäßen Verfahren in einem Funktionsblock 5 vorgebbare Informationen in den ungenutzten Speicherbereich des Programmspeichers lokatiert. Die vorgebbaren Informationen bewirken ein Zurücksetzen des Steuergerätes, wenn sie auf dem Rechengerät, insbesondere auf dem

Mikroprozessor oder CPU, abgearbeitet werden. Es ist aber auch denkbar, dass durch Abarbeiten der vorgebbaren Informationen auf dem Rechengerät zunächst eine Interrupt-Service-Routine oder eine Fehlerbehandlungs-Routine aufgerufen wird. Am Ende der Routine kann das Steuergerät dann bspw. durch einen Software-Reset zurückgesetzt und neu hochgefahren werden. Das Steuergerät befindet sich dann in einem definierten, voll funktionsfähigen Zustand und kann mit der Abarbeitung des Computerprogramms fortfahren.

Die vorgebbaren Informationen können lediglich in einem Teil der ungenutzten Speicherbereiche des Programmspeichers abgelegt werden. Vorzugsweise werden die vorgebbaren Informationen jedoch in allen ungenutzten Speicherbereichen des Programmspeichers abgelegt. Des Weiteren ist es möglich, dass die vorgebbaren Informationen lediglich in einem Teil eines ungenutzten Speicherbereiches, bspw. am Ende des ungenutzten Speicherbereiches, abgelegt werden. Dabei muss jedoch sichergestellt werden, dass der Teil des ungenutzten Speicherbereiches, in dem die vorgebbaren Informationen nicht abgelegt sind, keine Sprünge und keine Endlosschleifen bewirkt, wenn der darin enthaltene Programmcode auf dem Rechengerät abgearbeitet wird. Vorzugsweise wird jedoch ein ungenutzter Speicherbereich des Programmspeichers mit den vorgebbaren Informationen vollständig aufgefüllt.

In einem Funktionsblock 6 des erfindungsgemäßen Verfahrens wird der im Funktionsblock 4 lokatierte Programmcode des Computerprogramms und werden die in dem Funktionsblock 5 lokatierten vorgebbaren Informationen in den entsprechenden Speicherbereichen des Programmspeichers abgelegt. In einem Funktionsblock 7 ist das erfindungsgemäße Verfahren beendet.

Gemäß einem alternativen erfindungsgemäßen Verfahren, das jedoch nicht in den Figuren dargestellt ist, werden zunächst sämtliche vorhandene Speicherbereiche des Programmspeichers mit den vorgebbaren Informationen gefüllt. Anschließend wird nur der aus dem Lokator gewonnene Programmcode des Computerprogramms in die entsprechenden Speicherbereiche darüber geschrieben. Dieses Verfahren hat den Vorteil, dass dann über alle Bereiche eine Checksumme gebildet werden kann.

In Figur 2 ist ein erfindungsgemäßes Steuergerät gemäß einer bevorzugten Ausführungsform in seiner Gesamtheit mit dem Bezugszeichen 10 bezeichnet. Das Steuergerät 10 dient bspw. zur Steuerung und/oder Regelung bestimmter Funktionalitäten in einem Kraftfahrzeug. Das Steuergerät 10 umfasst einen CPU (Central Processing Unit)-Baustein 30 und einen separaten IC (Integrated Circuit)-Baustein 31, auf dem ein externer Flash-Speicher 20 angeordnet ist. Der CPU-Baustein 30 umfasst ein Rechengerät 11, das bspw. als ein Mikroprozessor oder als eine CPU ausgebildet ist. Das Rechengerät 11 steht über eine erste Datenverbindung 12 mit einem schnellen Arbeitspeicher 13 in Verbindung, der als ein statisches oder dynamisches RAM (Random-Access-Memory) ausgebildet ist. Über eine zweite Datenverbindung 14 steht das Rechengerät 11 mit einem internen Flash-Speicher 15 und einem ROM (Read-Only-Memory) 16 in Verbindung. Das ROM 16 ist ein Festwertspeicher, in dem bspw. das sogenannte BIOS (Basic Input Output System) abgelegt ist.

Über eine dritte Datenverbindung 17 und einen Kommunikationskontroller 18 steht das Rechengerät 11 mit einem Datenbus 19 in Verbindung. An den Datenbus 19 ist bspw. der externe Flash-Speicher 20 angeschlossen, so dass

das Rechengerät 11 über den Datenbus 19 auf den externen Flash-Speicher 20 zugreifen kann. Der interne Flash-Speicher 15 und der externe Flash-Speicher 20 bilden den Programmspeicher, in dem das Computerprogramm 21 in bestimmten Speicherbereichen und in den übrigen ungenutzten Speicherbereichen die vorgebbaren Informationen 22 abgelegt sind. Über eine vierte Datenverbindung 23 ist das Rechengerät 11 mit einer Schnittstelle 24 verbunden, über die eine Vorrichtung 25 zum Ablegen des Computerprogramms 21 und der vorgebbaren Informationen 22 in den dafür vorgesehenen Speicherbereichen des Programmspeichers angeschlossen werden kann.

Anhand des internen Flash-Speichers 15 wird der Stand der Technik beschrieben, bei dem während der Ausführung des Computerprogramms 21 (Pfeil 26) fälschlicherweise in den nachfolgenden ungenutzten Speicherbereich des Flash-Speichers 15 gesprungen wird (Pfeil 27). Das Rechengerät 11 liest Programmbefehle aus den ungenutzten Speicherbereich ein und führt diese aus (Pfeil 28). Da es sich bei den aus dem ungenutzten Speicherbereich eingelesenen Programmbefehlen um vorgebbare Informationen handelt, durch welche das Steuergerät 10 in einen definierten Zustand überführt wird, besteht bei der vorliegenden Erfindung keine Gefahr, dass das Computerprogramm 21 in eine unkontrollierte Endlosschleife oder das Steuergerät 10 in einen undefinierten und damit irregulären Zustand gelangt. Der definierte Zustand des Steuergerätes 10 kann bspw. durch ein Software-Reset erzielt werden. Dazu springt das Rechengerät 11 durch die Abarbeitung der Programmbefehle (Pfeil 28) veranlasst an eine bestimmte RESET-Adresse (Pfeil 29) eines Speicherbereichs des Programmspeichers 15, 20, in dem das Computerprogramm 21 abgelegt ist. Danach läuft das Computerprogramm 21 wieder hoch und die

Abarbeitung des Computerprogramms 21 beginnt wieder von vorne.

Der definierte Zustand des Steuergeräts 10 kann aber auch durch die Abarbeitung einer Interrupt-Service-Routine oder einer Fehlerbehandlungs-Routine erzielt werden. Dazu springt das Rechengerät 11 durch die Abarbeitung der Programmbefehle (Pfeil 28) veranlasst an eine bestimmte Speicher-Adresse eines Speicherbereichs des Programmspeichers 15, 20, in dem das Computerprogramm 21 abgelegt ist. Diese Speicher-Adresse entspricht dem Anfang der Interrupt-Service-Routine oder der Fehlerbehandlungs-Routine. Nach Abarbeitung der Interrupt-Service-Routine oder der Fehlerbehandlungs-Routine kann das Rechengerät 11 an die RESET-Adresse springen, um das Steuergerät 10 zurückzusetzen.

Die in den ungenutzten Speicherbereichen abgelegten vorgebbaren Informationen sind bspw. als ein Programmcode in einem Hexadezimal-Format, sogenannter Hexadezimal-Code, ausgebildet. Zum Auffüllen der ungenutzten Speicherbereiche des Programmspeichers 15, 20 kann die Vorrichtung 25 bspw. einen Hexadezimal-Editor umfassen. Der für die vorgebbaren Informationen verwendete Hexadezimal-Code sollte eine der folgenden Möglichkeiten erfüllen:

- den gesamten ungenutzten Speicherbereich mit mindestens einem Programmbefehl auffüllen, der gezielt ein Zurücksetzen des Steuergerätes 10 (sogenannter Reset) auslöst. Bei einem Microcontroller vom Typ 80C166 der Firma Siemens entspricht dies bspw. einem Befehl SRST oder Illegal Opcode.

- den gesamten ungenutzten Speicherbereich mit

mindestens einem Programmbefehl füllen, durch den in eine Interrupt-Service-Routine gesprungen wird.

- den gesamten ungenutzten Speicherbereich mit mindestens einem Programmbefehl füllen, durch den in eine spezielle Fehlerbehandlungs-Routine gesprungen wird.

- nur am Ende eines ungenutzten Speicherbereichs und ggf. zusätzlich in regelmäßigen Abständen, z. B. alle 512 Byte, innerhalb des ungenutzten Speicherbereiches mindestens einen Programmbefehl entsprechend den obigen drei Möglichkeiten implementieren. Das setzt allerdings voraus, dass die sonstigen Programmbefehle in dem ungenutzten Speicherbereich keine Sprünge und keine Endlosschleifen bewirken.

Am Ende der Interrupt-Service-Routine und am Ende der Fehlerbehandlungs-Routine sollte das Steuergerät bspw. per Software zurückgesetzt werden. Der Software-Reset hat ein erneutes Hochfahren des Steuergerätes 10 zur Folge. Sowohl bei der Interrupt-Service-Routine als auch bei der Fehlerbehandlungs-Routine können bspw. Vergangenheitsinformationen über den genauen Ort des Auftretens und die Vergangenheit festgehalten werden, z.B. Rücksprungadresse. Aus diesen Informationen können Rückschlüsse bspw. auf die Häufigkeit des Auftretens eines Fehlers, gemacht werden.

R.302905

28.06.2002
Robert Bosch GmbH, 70442 Stuttgart

Ansprüche

1. Verfahren zum Ablegen eines Computerprogramms (21) in einen Programmspeicher (15, 20) eines Steuergeräts (10), wobei das Computerprogramm (21) nach vorgebbaren Regeln in bestimmten Speicherbereichen des Programmspeichers (15, 20) abgelegt wird, **dadurch gekennzeichnet**, dass in ungenutzten Speicherbereichen des Programmspeichers (15, 20), in denen das Computerprogramm (21) nicht abgelegt wird, vorgebbare Informationen (22) abgelegt werden, durch die das Steuergerät (10) in einen definierten Zustand überführt wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das Steuergerät (10) durch Abarbeiten der vorgebbaren Informationen (22) auf einem Rechengerät (11), insbesondere auf einem Mikroprozessor, des Steuergeräts (10) zurückgesetzt wird.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass durch Abarbeiten der vorgebbaren Informationen (22) auf einem Rechengerät (11), insbesondere auf einem Mikroprozessor, des Steuergeräts (10) eine Interrupt-Service-Routine aufgerufen wird.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass durch Abarbeiten der vorgebbaren Informationen (22) auf einem Rechengerät (11), insbesondere auf einem Mikroprozessor, des Steuergeräts (10) eine Fehlerbehandlungs-Routine aufgerufen wird.

5. Verfahren nach Anspruch 3 oder 4, dadurch gekennzeichnet, dass das Steuergerät (10) am Ende der Routine zurückgesetzt wird.

6. Verfahren nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass die vorgebbaren Informationen (22) in allen ungenutzten Speicherbereichen des Programmspeichers (15, 20) abgelegt werden.

7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass wenigstens ein ungenutzter Speicherbereich des Programmspeichers (15, 20) mit den vorgebbaren Informationen (22) vollständig aufgefüllt wird.

8. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass die vorgebbaren Informationen (22) in vorgebbaren Abständen in wenigstens einem ungenutzten Speicherbereich des Programmspeichers (15, 20) abgelegt werden, wobei der Teil des ungenutzten Speicherbereichs, in dem die vorgebbaren Informationen (22) nicht abgelegt werden, keine Sprünge und keine Endlosschleifen bewirkt.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, dass die vorgebbaren Informationen (22) in regelmäßigen Abständen in dem wenigstens einen ungenutzten Speicherbereich des Programmspeichers (15, 20) abgelegt werden.

10. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass die vorgebbaren Informationen (22) am

Ende wenigstens eines ungenutzten Speicherbereichs des Programmspeichers (15, 20) abgelegt werden, wobei der Teil des ungenutzten Speicherbereichs, in dem die vorgebbaren Informationen (22) nicht abgelegt werden, keine Sprünge und keine Endlosschleifen bewirkt.

11. Vorrichtung (25) zum Ablegen eines Computerprogramms (21) in einen Programmspeicher (15, 20) eines Steuergeräts (10), wobei die Vorrichtung (25) erste Mittel zum Ablegen des Computerprogramms (21) nach vorgebbaren Regeln in bestimmten Speicherbereichen des Programmspeichers (15, 20) aufweist, **dadurch gekennzeichnet**, dass die Vorrichtung (25) zweite Mittel zum Ablegen von vorgebbaren Informationen (22), welche das Steuergerät (10) in einen definierten Zustand überführen, in ungenutzte Speicherbereiche des Programmspeichers (15, 20) aufweist, in denen die ersten Mittel das Computerprogramm (21) nicht abgelegt haben.

12. Vorrichtung (25) nach Anspruch 11, dadurch gekennzeichnet, dass die zweiten Mittel als ein Hexadezimal-Editor ausgebildet sind.

13. Vorrichtung (25) nach Anspruch 11 oder 12, dadurch gekennzeichnet, dass die Vorrichtung (25) Mittel zur Ausführung eines Verfahrens nach einem der Ansprüche 2 bis 10 aufweist.

14. Steuergerät (10) mit einem Rechengerät (11), insbesondere einem Mikroprozessor, und einem Programmspeicher (15, 20), auf dem ein Computerprogramm (21) nach vorgebbaren Regeln in bestimmten Speicherbereichen des Programmspeichers (15, 20) abgelegt ist, **dadurch gekennzeichnet**, dass in ungenutzten Speicherbereichen des Programmspeichers (15, 20), in denen das Computerprogramm (21) nicht abgelegt ist, vorgebbare

Informationen (22) abgelegt sind, durch die das Steuergerät (10) in einen definierten Zustand überführbar ist.

15. Steuergerät (10) nach Anspruch 14, dadurch gekennzeichnet, dass die vorgebbaren Informationen (22) nach einem Verfahren nach einem der Ansprüche 2 bis 10 in den ungenutzten Speicherbereichen des Programmspeichers (15, 20) abgelegt sind.

R.302905

28.06.2002

Verfahren und Vorrichtung zum Ablegen eines Computerprogramms in einen Programmspeicher eines Steuergeräts

Zusammenfassung

Die Erfindung betrifft ein Verfahren und eine Vorrichtung (25) zum Ablegen eines Computerprogramms (21) in einen Programmspeicher (15, 20) eines Steuergeräts (10). Das Computerprogramm (21) wird nach vorgebbaren Regeln in bestimmten Speicherbereichen des Programmspeichers (15, 20) abgelegt. Um einen fälschlicherweise erfolgten Sprung in einen ungenutzten Speicherbereich des Programmspeichers (15, 20), in dem das Computerprogramm (21) nicht abgelegt ist, möglichst schnell zu erkennen und um zu verhindern, dass sich das Steuergerät (10) in einem irregulären Zustand befindet, wird vorgeschlagen, dass in den ungenutzten Speicherbereichen des Programmspeichers (15, 20), in denen das Computerprogramm (21) nicht abgelegt wird, vorgebbare Informationen (22) abgelegt werden, durch die das Steuergerät (10) in einen definierten Zustand überführt wird. (Figur 2)

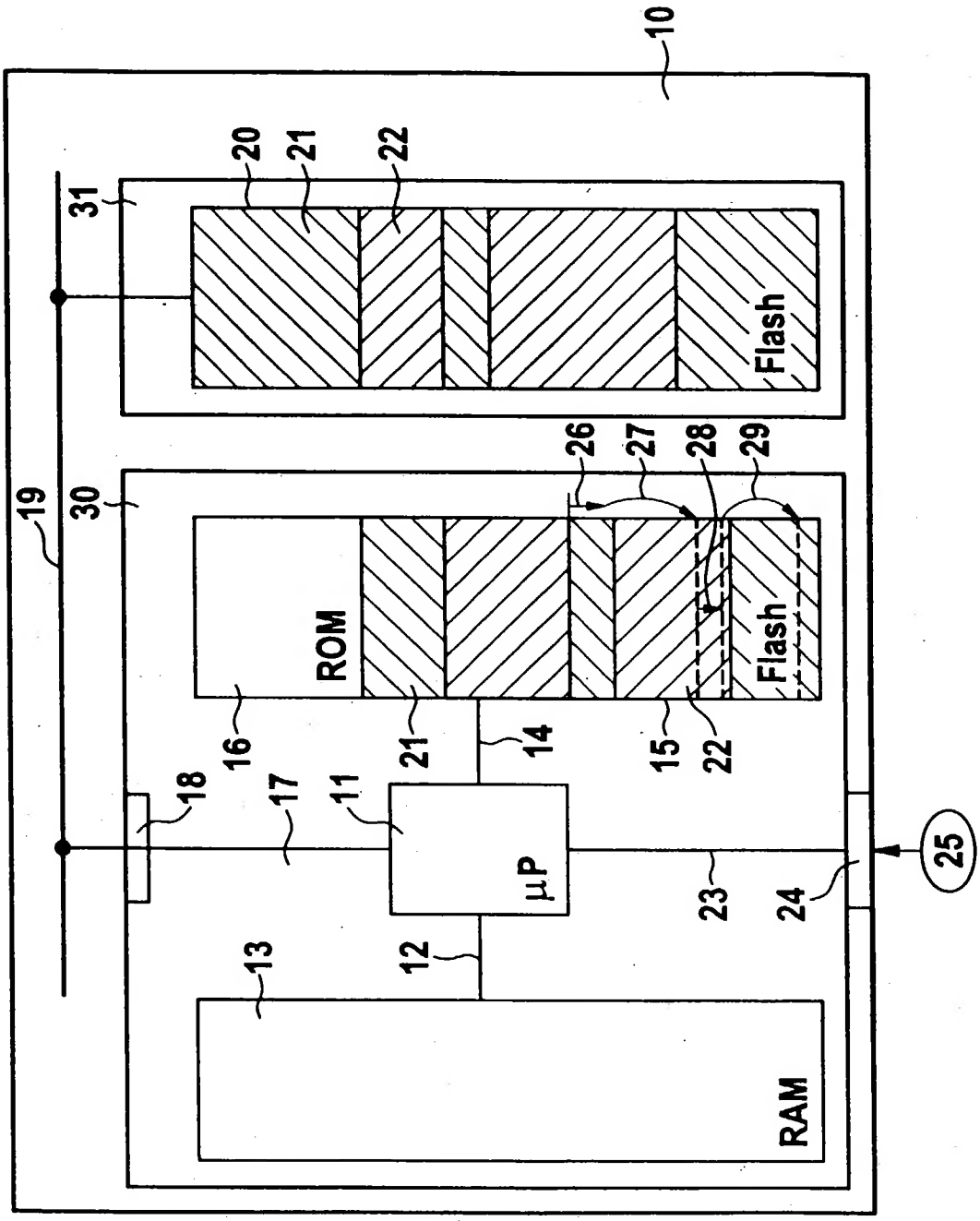
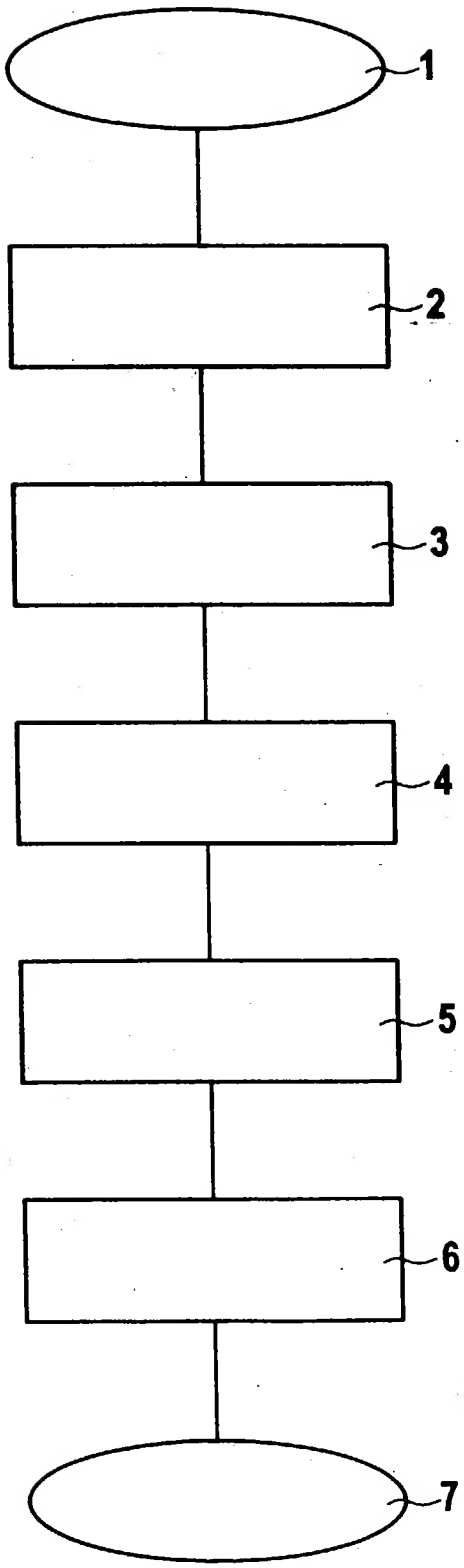

10
20
21
22
31
Flash
26
27
28
29
19
30
ROM
Flash
16
21
14
15
22
18
17
11
µP
25
23
24
13
12
RAM

Fig. 2

Fig. 1



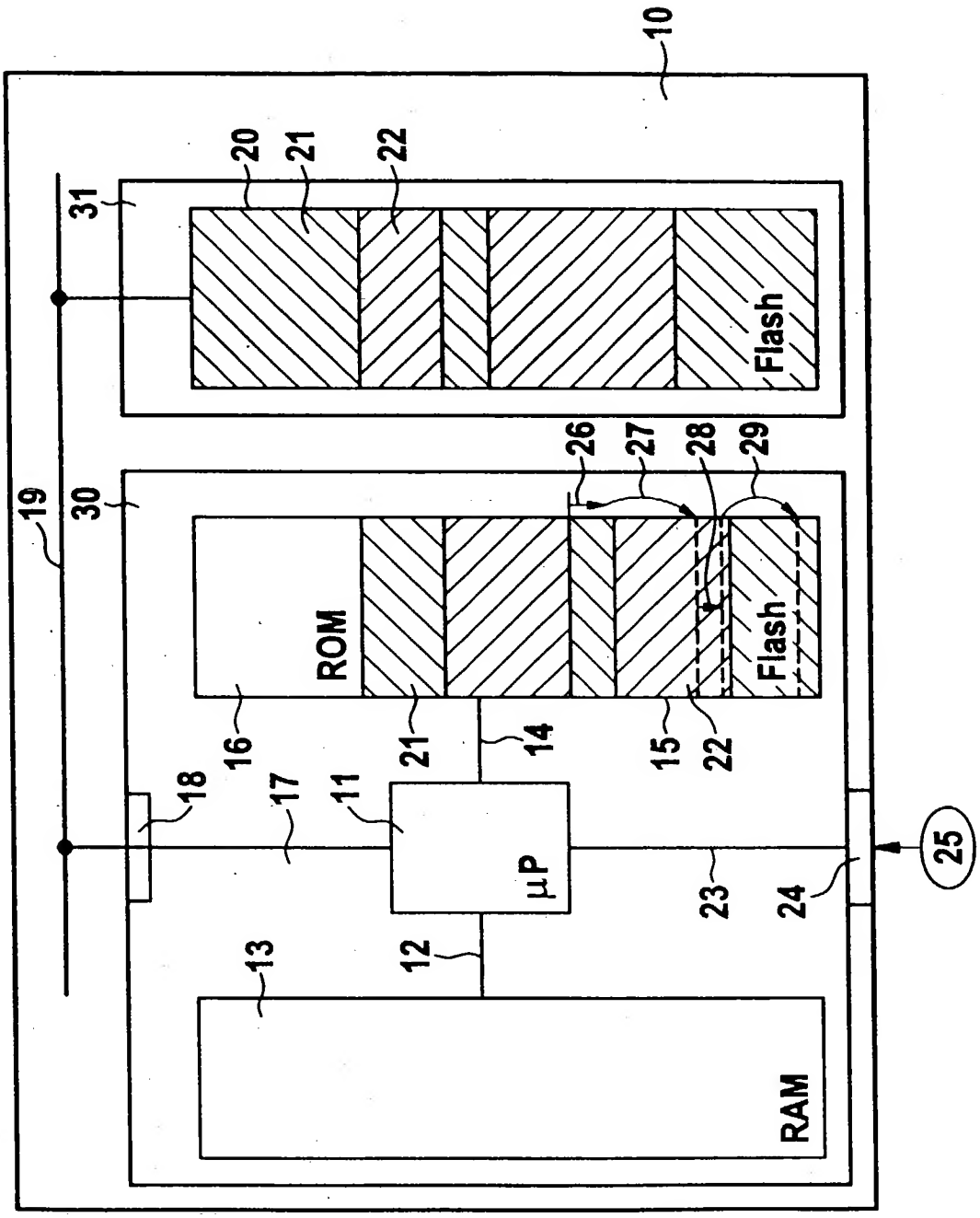

10
20
21
22
31
Flash
26
27
28
29
19
30
ROM
Flash
16
21
14
15
22
18
17
11
μP
25
23
24
13
12
RAM

Fig. 2